International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-4, November 2019

Mispronunciation Detection for Spoken Isolated Words using Segmentation and Classification under Low Resource Conditions for Kannada Language

Savitha Murthy, Pragnya Suresh, Preet Shah, Dinkar Sitaram

Abstract: ***Relocation makes it inevitable for a person to learn the local pronunciations correctly. With the advent of mobile phones, language learning can be made easy and flexible. Our research involves Kannada Kali, a mobile and cloud based application that is being developed to assist people in learning the correct pronunciations of Kannada (a language spoken in India). Automatic Speech Recognition systems which are used to aid pronunciation training require to be trained on sufficient amount of spoken target language data. Since collecting such data in not easy, the objective of our research is to detect mispronounced segments of words with minimal data. When there is scarcity of data, a comparative approach where spoken word segments are compared with the canonical pronunciations is more effective for detecting anomaly in pronunciation. Since syllables are basic independent units of pronunciation, the spoken words are segmented into syllables for effective comparison and feedback. We propose an unsupervised segmentation method called Spectrogram Formant Contour Analysis that detects syllable boundaries by analysing the change in contours of the formants in the spoken word spectrograms. The task of mispronunciation detection is more effective when the application can identify the actual syllable pronounced and communicate the correct pronunciation to the user. For the purpose of syllable classification, our method employs a novel approach where a model is trained on phonemes and given syllables as input for identification. Our study includes comparing the performance of three machine learning algorithms, namely, Convolution Neural Network, Support Vector Machines and K-Nearest Neighbours on the task of identifying phonemes when they are trained on minimal data. The accuracy of KNN on phoneme classification was the best with 80% for clean and 60% for noisy data. In case of our initial results on syllable classification for Kannada Kali, Support Vector Machines gave the highest accuracy of almost 30%.***

Keywords**:** ***Mispronunciation Detection, Kannada isolated words, syllable segmentation, Spectrogram Format Contour Analysis.***

Manuscript published on November 30, 2019.
* Correspondence Author
Savitha Murthy*, Department of CSE, PES University, Bangalore, India.
Pragnya Suresh, Department of CSE, PES University, Bangalore, India.
Preet Shah, Department of CSE, PES University, Bangalore, India.
Dinkar Sitaram, Department of CSE, PES University, Bangalore, India.

I. INTRODUCTION

India is known for its diversity. Multiple cities have gained prominence in various domains. Consequently, more and more people are relocating to these places. For ease of communication, it has become necessary for them to speak the local tongue. Smart phones have become ubiquitous and a mobile application would allow people to learn the language easily. To enable them to learn the correct pronunciation of the word, it is essential that the application points out specific segments of the word that the user mispronounced so that the learner can improve his or her pronunciation.

Computers and machine learning have made learning a new language more easy and flexible for a non-native (L2) learner. There are two systems that aid in learning to speak a language. While Computer Aided Language Learning (CALL) systems act as a guide for language learning including linguistic features and grammar, Computer Aided Training (CAPT) systems emphasize on the pronunciation aspects of the learner. There are two parts to a CAPT system - one is assigning a pronunciation evaluation score to the learner's pronunciation and another part in to specifically indicate where the pronunciation went wrong. Learning to speak is more effective when the system is able to indicate where the user's pronunciation deviates from the canonical pronunciation. Mispronunciation detection is possible with the help of traditional Automatic Speech Recognition (ASR) systems that are trained is large amount of language data and detect mispronunciations based on the posterior probabilities or Goodness of Pronunciation (GOP) scores at word, sub-word or phone level. ASR systems typically require lexicons for specifying pronunciations (this is done by a language expert), long hours of audio recordings and corresponding transcripts and language model that models the grammar of the target language. Additionally CAPT systems prefer ASR systems that are trained on both native (L1) and non-native (L2) recordings.

II. RELATED WORK

Mispronunciation detection is one of the core components of CAPT systems.

Earlier work on mispronunciation detection made use of GMM (Gaussian Mixture Model)-HMM (Hidden Markov Model) based models and recently DNN (Deep Neural Network) for obtaining the posterior probabilities of the pronunciations at phoneme level. Reference [1] achieves pronunciation error detection at phoneme level for non-native Arabic. With deep learning proving its efficacy, the work in [2] demonstrates that a DNN based pronunciation evaluation system is more efficient than a GMM-HMM model as they make use of frame level posterior probabilities and do not require a decoding lattice. They make use of likelihood, log likelihood and frame level posterior probabilities scores for pronunciation evaluation. They further their research with the inclusion of tonal information into the DNN model [3]. The work in [4] focuses on improving mispronunciation detection using DNN. Work in [5] involves development of a cross-platform mobile application for pronunciation training where the backend is hosted on a cloud. They make use of a traditional ASR system to detect the place of mispronunciations and display them on the mobile application.There has been research to minimize dependency on large speech corpus for pronunciation training. For one, pronunciation training is more effective when trained on non-native speech data. Such data is difficult to gather. Research to eliminate the need for non-native training data is conducted by [6] and [7]. The research in [7] is on eliminating the need for non-native training data. They map the pronunciations between native and target languages. They train a multi-lingual DNN to model the articulatory features of of non-native speech. [6] have used a similar approach. They evaluate pronunciations of Indian English using native English and native Hindi speech. They train a GMM-HMM based acoustic model. Reference [8] uses the concept of anti-phones to detect substitutions in pronunciation and a filler model to detect insertions. This prevents the need for error pattern modeling.

Another approach to evaluate speech is GOP (Goodness of Pronunciation) scores with Logistic Regression (LR) proposed by [9]. A variant called forced-aligned GOP (F-GOP) was used. The LR classifier on top of the F-GOP algorithm outperformed the baseline due to the possibility to add informative features as input to the classifier. Mispronunciation at the phoneme level has been studied before using DNN models and Long Short Term Memory (LSTM) [10]. Various methods such as log-likelihood ratio [11] and GOP [12] were adopted previously. To obtain more accurate information about mispronunciation, they used an LSTM embedded pronunciation vector. Reference [13] describes six different models to detect errors in speech. Using phonological features, a computer aided pronunciation training system was presented in [14]. A multi-task deep neural network model was used to estimate HMM state probabilities. Their proposed system could accurately detect mispronunciations.Detecting mispronunciations has also been done using automated tools. One such paper [15] uses Kaldi speech recognition toolkit to detect miscues in speech based on two features to detect incorrect speech - the target phonemes found in the speech output of user and automatic speech recognition confidence scores providing speech accuracy. The paper uses phoneme-level statistics and also does prosody analysis by prosody feature extraction. The work in [16] estimates acoustic models by leveraging deep neural network. All the above methods mentioned above require sufficient amount of speech recordings for training the models for efficient mispronunciation detection.

Comparison techniques have been used for pronunciation evaluation without involving a speech corpus [17][18]. Research in [17][18] implements a comparison based pronunciation evaluation which eliminates the need for traditional ASR systems with enough training data in both native and non-native speech. They compare the spoken sentences of a teacher and student using various computations on Dynamic Time Warping (DTW) of the two utterances. They train a Support Vector Machines (SVM) classifier on phoneme based and word based features to evaluate a student's speech.The objective of our research is to obtain effective mispronunciation detection without using an ASR. This is achieved by segmenting the spoken words into syllables. We propose Spectrogram Formant Contour Analysis (SFCA) method for obtaining speech segmentation. Speech segmentation is performed as a primary step in many speech related applications including ASR where segments (normally phonemes) are aligned with the transcripts for training. The objective of our research is to achieve mispronunciation detection with minimal data. We adapt a comparative template based approach where the learner's pronunciation is compared with a canonical pronunciation of a native speaker. This is more effective when segments of speech can be compared and hence the spoken words are segmented into syllables. Speech segmentation of isolated words refers to the process of identifying the boundaries between syllables, or phonemes in spoken natural languages. The basic phonetic unit can be a phoneme or a syllable based on the language. Our work considers syllable as the basic unit since Indian native languages are syllable-time based and syllables form independent units of pronunciation. Identifying the syllable represented by a speech segment is an important step in mispronunciation detection. The segmented syllables are classified in order to convey the right pronunciation to the learner in case of a mispronunciation. The performance of Convolutional Neural Network (CNN), Support Vector Machines (SVM) and K-Nearest Neighbours (KNN) are evaluated for phoneme classification. Since SVM and KNN perform better in noisy conditions they are futher evaluated for syllable classification. The results obtained depict that SVM performs better than KNN under noisy conditions. This is essential since the recordings from a mobile application are bound to contain background noise.

III. OUR APPROACH

Mispronunciation detection is more effective when a pronunciation training system can indicate to the learner the part of the word that has been spoken incorrectly. A step towards this is to first identify what the learner has pronounced. Our method for mispronunciation detection involves two parts – first is to segment the spoken words into syllables and second is to classify the syllable. Segmentation is achieved through a novel unsupervised method called Spectrogram Formant Contour Analysis (SFCA) described in section III A.

The validity of this method is verified with aligning the segments with a well established Peak Valley Detection method. Syllable classification is achieved by first training a model on individual phonemes. After segmenting a spoken word using FCA, syllables are given as input to the model. The most probable vowel and consonant identified by the model (trained on phonemes) are combined to determine the syllable. The performance of CNN, SVM and KNN are compared for phoneme classification with minimal data; SVM and KNN are compared for syllable classification from the recordings of Kannada Kali application. The details of classification are described in section III B.

A. *Unsupervised Audio Segmentation into Syllables*

1) Peak Valley Detection - Baseline

Peak Valley detection is a well established method for segmentation [19][20][21][22]. Since the focus of our research is segmenting spoken words that can be spoken under noisy conditions using a mobile application, we further apply smoothing on the obtained peaks and valleys to handle noisy speech. This method is used as a baseline to establish the validity of SFCA proposed in this work.

As a pre-processing step, the spectrogram of the audio signal $s[n]$ (where n is length of the signal) obtained and the window length and maximum frequency is set. The intensity values across audio frequency axis at every time frame are added to obtain the amplitude $a[n]$ from $s[n]$ as in equation (1),

$$a[n] = \sum_{i=0}^{n}[I_i], I \in s[n] \quad (1)$$

Smoothing is applied on a[m] which uses convolution of a scaled window (flat window in this case) with a[m] as,

$$c[n] = (a * v)[n] = \sum_{m=-\infty}^{\infty} a[m]\, v[n-m] \quad (2)$$

Smoothing peaks and valleys of noisy speech:

The peaks p[l], where l is the number of peaks and valleys v[q], where q is the number of valleys were obtained. Numerous peaks and valleys were initially identified because of noisy speech. Smoothing was applied to eliminate peaks and valleys that were closer than empirically determined thresholds. The steps followed to smooth the peaks and valleys are as follows:

1. Delete any peak for $n \leq 15$
2. Delete any peak if $[i] \leq th_{height}$, where th_{height} is threshold for the height of the peak.
3. Delete the shorter peak if $p[i] - p[i+1] \leq th_{peak\_dist}$, where th_{peak_dist} is the threshold for the distance between peaks set.
4. Delete a valley if $p[i] - v[i+1] \leq th_{dist}$ and height of $[i] > 1.3$, where th_{dist} is the distance threshold between peak and following valley.
5. Delete the smaller valley if $v[i] - v[i+1] \leq th_{valley\_dist}$, where th_{valley_dist} is the distance between two valleys.

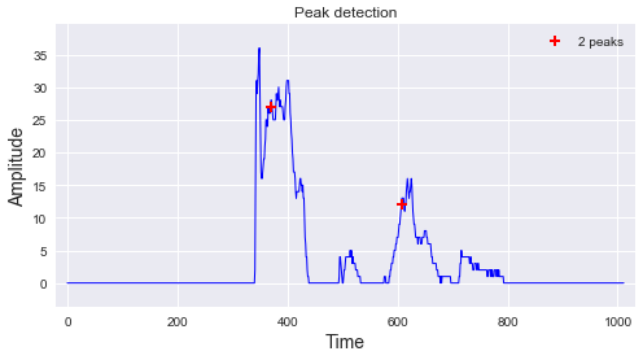
The peaks array is processed to filter out peaks that are detected in the noise region and also those peaks whose heights are less than th_{height} of 0.85 (step 2). If the waveform is erratic, too many peaks that are very close to each other may be identified. Multiple peaks detected between two syllable boundaries will result in the formation of too many segments. In order to prevent this, if the distance between two consecutive peak indices is less than th_{peak_dist}, then the larger of the two peaks is retained and the smaller one is eliminated (step 3). If the valley is detected very close to a peak, and if the absolute difference between the heights of the peak and valley is less than a th_{dist} set at 1.7(given that the height of the peak is greater than 1.3) ,then the valley is removed (step 4). If the difference between the height of the peak and the corresponding valley is very less, it indicates that the valley represents a slight dip in the waveform and will not mark a syllable boundary. Hence, such valleys are removed from the list. If multiple valleys are detected very close to each other due to irregularities in the waveform (due to noise), the syllable will be divided into further segments. In order to avoid this, if the difference between two valley indices is less than th_{valley_dist}, then the smaller of the two is retained (step 5). Silence regions are smoothened by eliminating any frivolous valleys/peaks detected. For each peak in the peaks list, the smallest valley that follows the peak (to mark the end of the syllable segment) is appended if the difference between the indices of the given peak and valley is greater than a threshold. If there is no such valley found, the largest element of valley list that precedes the current peak (to mark the beginning of the syllable segment) is appended to the final syllable boundary list. Fig. 1 depicts the smoothing and filtering of peaks and valleys for the Kannada word '/ba:/ /tu/ /ko:/ /ḷi/' (which means duck) which is segmented into four syllables.

2) Spectrogram Formant Contour Analysis

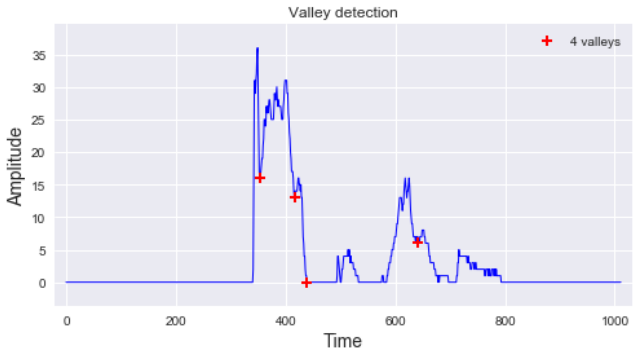
It was observed found that the patterns of the spectrogram were unique for each sound and change in pattern can be an indication of change in sound aiding segmentation. The objective of our method is to determine the points of segmentation by studying the change in contour patterns of the formants F1, F2 and F3, the details of which are given below. Formants are regions of high acoustic energy in a spectrogram.

We obtain the spectrogram for the audio (isolated spoken word in this case) and determine the formants F1, F2 and F3. We obtain log spectrogram and empirically determine the threshold value for considering the start of the spoken word given by equation (3). Only regions of the audio signal with intensities higher than noise threshold,th_{noise} are considered for further processing.

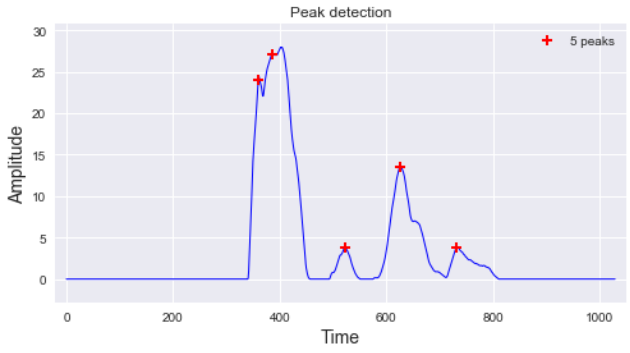
$$th_{noise} = {\left(\sum_{i=0}^{n} \log(s[i])\right)}/{n} + \sum_{i=0}^{n} \log(s[i]) + \sqrt{\sum_{i=0}^{n} \log\left(s[i]\right)} \quad (3)$$



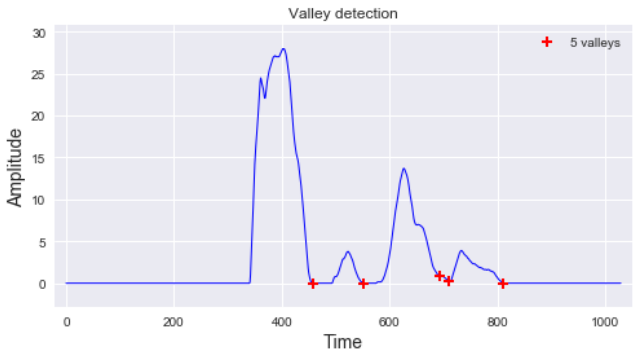
(a)



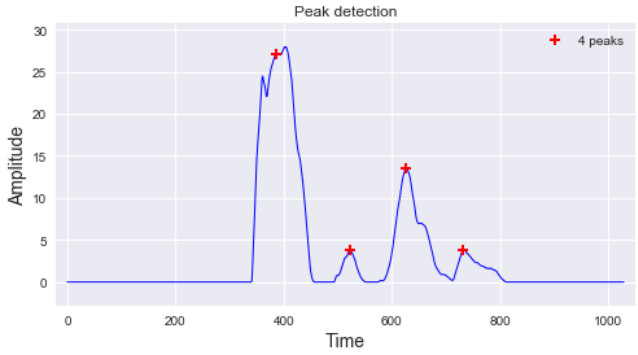
(b)



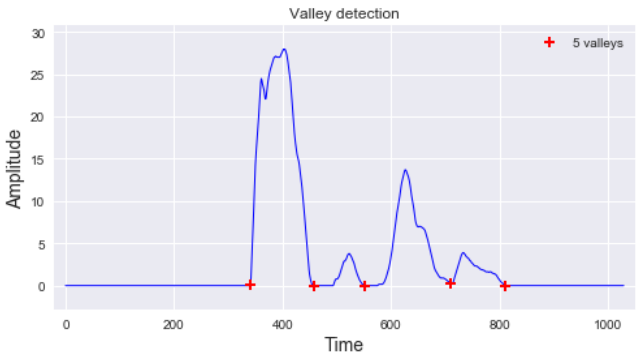
(c)



(d)



(e)



(f)

Syllable boundaries are determined by analysing the contours of the formants in the audio spectogram. The changes in the contour of the three formants, F1, F2 and F3 are considered to mark the syllable boundaries as depicted in Fig. 2. Different weights are assigned to changes in respective formant in order of prominence (highest wieght to F1 and lowest to F3). The processing steps are summarized in the algorithm listed in Fig 3. For the purpose of analysis, we consider three row groups - r1, r2 and r3 near each of the formant. Each group is formed by mean values around (one below and one above) each formant as specified in steps 2, 3 and 4. Thus at each point in time, we compute a total of nine values - 3 values (means) around each formant. To determine the syllable boundaries, we consider a window of three columns of the spectrogram at a time (steps 8, 9 and 10) with a stride of one. th_{col_dist} is the allowable absolute difference between corresponding values of two columns in the window. Scores are computed based on the absolute differences as specified in steps 14 to 24 between first column and second column formant groups. col_dist_{12} maintains the score for the differences between formant values of first column with that of second column while col_dist_{13} maintains the score for the differences between formant values of first column formant and third column. Values of $k = 1,2,3$ are for F1 group, $k = 4,5,6$ are for F2 group and $k = 7,8,9$ are for F3 group. The weightage assigned are 0.8, 0.5 and 0.3 for F1, F2 and F3 groups respectively. Higher priority is given to differences between first formant (F1) groups and lowest to the differences between third formant (F3) groups. This is because first formant is more predominant than other formants. Everytime two column values are less than a threshold (that means they are similar) the scores col_dist_{12} or col_dist_{13} are incremented. The accumulated scores of col_dist_{12} and col_dist_{13} for a window are compared with th_{bndry}. A higher value indicates similar columns. Similar computations are repeated for second column and third column formant groups with a threshold of col_dist_{23} (step 25).

The possible candidates for syllable boundaries, represented as magenta colored points in Fig. 2, are determined based on the following conditions:

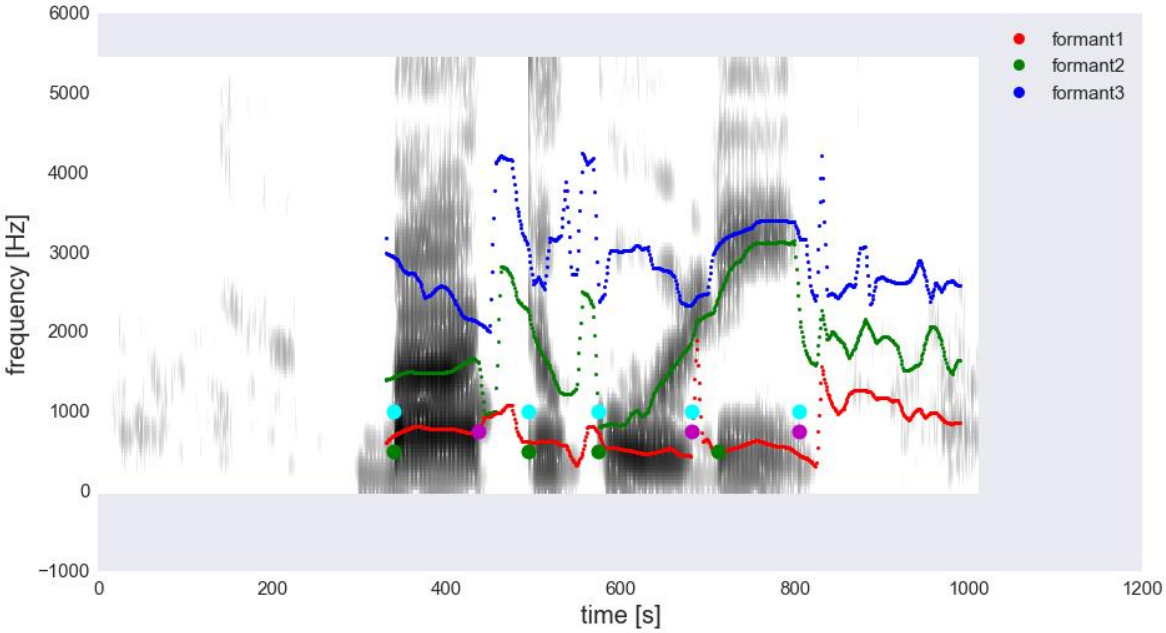


Fig. 2. Spectrogram Formant Contours with possible syllable boundaries

Algorithm 1: Segmenation based on Spectrogram Formant Contour Analysis

```
while i ≤ n do
    r1 ← [(Σ(f1[i] − 1, ..., f1[i] + 2))/4, (Σ(f1[i] − 4, ..., f1[i] − 1))/4, (Σ(f1[i] + 2, ..., f1[i] + 5))/4]
    r2 ← [(Σ(f2[i] − 1, ..., f2[i] + 2))/4, (Σ(f2[i] − 4, ..., f2[i] − 1))/4, (Σ(f2[i] + 2, ..., f2[i] + 5))/4]
    r3 ← [(Σ(f3[i] − 1, ..., f3[i] + 2))/4, (Σ(f3[i] − 4, ..., f3[i] − 1))/4, (Σ(f3[i] + 2, ..., f3[i] + 5))/4]
    formants ← (r1,r2,r3)
end
for i ← 0 to n do
    c1 ← formants[i]
    c2 ← formants[i + 1]
    c3 ← formants[i + 2]
    if Σ(a3) ≤ val then
        sil_arr ← formant[i]
    end
    wieghtage = [0.8, 0.5, 0.3] for k ← 1 to 9 do
        if |c1[k] − c2[k]| ≤ th_col_dist then
            col_dist12 ← col_dist12 + ⌊k/3⌋
        end
        if |c1[k] − c3[k]| ≤ th_col_dist then
            col_dist13 ← col_dist13 + ⌊k/3⌋
        end
    end
    if col_dist12 ≥ th_bndry and col_dist13 ≤ th_bndry then
        syllable_boundary ← i
    end
    Repeat steps 15 to 24 for c2[k] and c3[k] with col_dist23asthreshold
end
```

Fig. 3. Algorithm for segmentation based on Spectrogram Contour Analysis

1. If $c1 \cong c2$ and $c1 \cong c3$, all the columns are part of the same syllable and no segmentation is needed.
2. If $c1 \neq c2$ and $c1 \neq c3$, the syllable boundary is set at $c1$.

3. if $c1 \cong c2$ and $c2 \neq c3$, the syllable boundary is set at $c2$.

Green dots in figure Fig. 2 represent possible candidates silences between syllables. sil_arr contains the column indices of the spectrogram with low intensity values (steps 11 to 13). Empirical thresholds are set to account for such frames that occur due to the low intensity periods during vibration of the vocal cords while speaking. sil_arr is pruned to ensure that the silence regions are long enough to be qualified as silence. The list of possible syllable boundaries is further pruned againt the list of silences to obtain a final list of syllable boundaries indicated as dots in cyan in Fig. 2.

B. *Mispronunciation Detection*

Mispronunciation detection is achieved by identifying the syllables from the segmented words and determining theie similarity with the syllables in the canonical pronunciation. Though syllables are independent units of pronunciation, one or more phoneme together forms a syllable. Phonemes are the basic pronunciation units with no variation. Hence, we first train different models on phonemes after manually segmenting the syllables. There are two steps involved – first is to train a model on individual phonemes and use the model to identify the phonemes constituting the syllable spoken.

Our experiments are conducted for a subset of 26 syllables of the Kannada alphabet consisting of 6 vowels and 14 consonants. Recordings of phonemes (clean samples) were obtained from six speakers of which four were used for training and two for testing. We also used phonemes after manually segmenting spoken Kannada words from Kannada Kali application (noisy samples). Table I lists the phonemes and syllables from Kannada language considered for our experiments.

Table – I: Subset of Kannada alphabets

Vowels (phonemes)	/a/, /a:/, /e/, /i/, /o/, /u/
Consonants (phonemes)	/b/, /ch/, /d/, /g/, /h/, /k/ /kh/, /l/, /m/, /n/, /r/, /s/, /v/
Syllables	/se:/, /ka:/, /ge/, /ba:/, /ko/, /da:/, /sa/, /va:/, /ma/, /li/, /ka/, /na/, /ba/, /ra:/, /chu/, /ddu/, /ki/, /na:/, /ne/, /di/, /kha/, /ga/, /ru/, /ma/, /ra/, /ha/

The audio segments (phonemes / syllables) are further converted to mel scale for classification (this is because mel scale represents the nature of human perception of sound). Using the triangular filter-bank helps to capture the energy at each critical band and gives a rough approximation of the spectrum shape, as well as smoothens the harmonic structure, thus, both reducing dimensions and noise due to vibration of the vocal tract. Mel spectrogram parameters employed are listed in Table II.

Table – II: Melspectrogram parameters

Window length	0.005
Time step	0.005
Position of first filter	100
Distance between filters	100
Maximum frequency	0

Exponential and power functions are applied to the mel scaled signal as given in equation (4) to prevent negative values and also magnify the differences between values.

$$\hat{s}[n] = e^{2*s[n]} \quad (4)$$

where s[n] is the mel-scaled spectrogram

The values obtained are then scaled to [0,1] and given as input to a model for training or classification.

1) *Phoneme classification*

a) *CNN classification - Baseline*

Since CNN are the most preferred for pattern recognition, CNN framework is used as a baseline for evaluating phoneme classification. The input features to the CNN framework are spectrogram values with zero padding to ensure uniform input dimension. We use one-hot encoding for phonemes as target labels. The model consists of two convolution layers(2D) with 16 filters of dimension (1x1) followed by a maxpool layer of size (2x2). This is followed by another convolution layer of size (2x2), a maxpool layer of size 2x2 and two fully connected layers - one with relu activation and another with softmax activation. The 'rmsprop' optimiser is used to alter weights in such a way that the loss function ('categorical_crossentropy') is minimised at each epoch. We train the model for 80 iterations to obtain the baseline accuracy.

b) *KNN – classification*

Four recordings of each syllable split into its phonemes are stacked and given to the knn for training. The number of neighbors is set to 4 with Euclidean distance between data points. We use ball_tree algorithm with a leaf size of 30. We use `distance' weight parameter which assigns weights proportional to the inverse of the distance from the query point.

c) *SVM – classification*

Since SVMs are effective for classification in high dimensional space especially when the data is less than the number of dimensions, our study involved SVM for classifying the phonemes. The SVM model used in the current implementation uses RBF kernel for nonlinear decision boundary with kernel coefficient as `scale' which uses the value of $n * \sigma(s[n])$. We use a penalty factor of 10 on the error for regularization to prevent over-fitting on less data. The decision function used is `one versus one' which is suitable for multi-class classification. The consonant samples had very few rows in comparison to the vowels since vowel duration is longer than a consonant. Hence class weights were given to increase the priority of the different consonant classes as listed in Table III.

Table – III: Consonant and vowel weights for SVM training

Consonants	Weights
/kh/, /l/, /m/, /n/, /r/,/s/,/v/	2.15
/b/, /ch/, /d/, /g/, /h/, /k1/	1.46
/a/, /a:/, /ae/, /e/, /o/, /u/	1

2) *Syllable classification*

The syllables obtained after applying SFCA method on 14 spoken Kannada isolated words from Kannada Kali application, were then given as input to SVM and KNN models trained on phonemes. When given a syllable the model identifies the phonemes that form the syllable. Fig. 4 depicts the identification of syllables using phoneme classification.

A two syllable word /bi//ḷi/ is split using SFCA. While testing, individual frames of the mel spectrogram are given as input. The bin with the highest number of points is identified spoken phoneme. Two such identified phonemes of the syllables (most populated bins) are combined to determine the syllable.

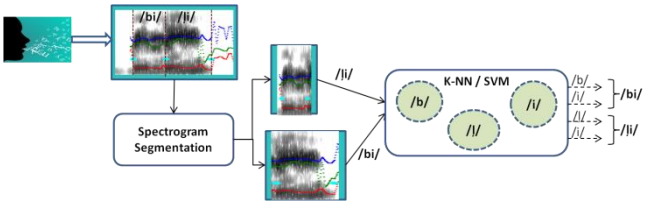


Fig. 4 KNN / SVM Syllable Classification

IV. RESULTS AND DISCUSSION

A. *Segmentation*

For validating SFCA method of syllable segmentation, a subset of Kannada words recorded as a part of the Kannada Kali application was segmented using both SFCA and Peak Valley Detection methods. Fig. 5 depicts the alignment of the syllable boundaries obtained using both the methods for four words with two, three, four and five syllables respectively. The yellow lines represent time alignment and the brown lines represent the syllable boundaries. All the syllable boundaries are approximately aligned except for the syllable boundaries in Fig. 5(b). The word /ma/ /lli/ /ge/ is split into three syllables using Peak Valley Detection method where SFCA splits it into four syllables. SFCA recognizes the /m/ sound at the beginning of a word as a separate syllable where is Fig. 5(d) /m/ is part of the syllable /mba/ of the word /ka/ /na/ /ka:/ /mba/ /ra/.

B. *Classification*

From the results presented in Table IV, phoneme classification accuracy of CNN is 50% using the clean samples. This value reduced when CNN was presented with data that had some noise component from the application. This indicates that CNN is not the best classifier while working with low resource data though it is robust to noise. SVM performed slightly better when tested with clean data but gave a lower accuracy than CNN with unclean data. KNN seems to perform the best when data is limited and also under noisy conditions. When manually trimmed phonemes were given, KNN outperformed the other models. Care was taken to exclude any noise or extra data frames. The samples used for training had a high impact on KNN accuracy.

Table – IV: Phoneme classification accuracy using clean and noisy audio

Method	Clean Audio	Noisy Audio
CNN	50%	50%
SVM	65%	35%
KNN	80%	60%

Table V depicts the number of vowels and consonants correctly classified by all the three models. SVM and KNN are comparable for vowel classification whereas KNN seems to outperform both CNN and SVM for consonant classification.Table VI lists the phonemes and syllables correctly identified for the Kannada Kali application. CNN is not used here as CNN is only used a baseline for phoneme classification. The comparison is only between SVM and KNN. The assumption was that since each phoneme has a unique pattern, the points of each phoneme would occupy a particular position in an n-dimensional space and these points could form a cluster to represent that phoneme. In both SVM and KNN methods, many of the consonant points have been placed in the vowels bins and this has caused the accuracy to drop a little. This is because vowels have higher values and any stray points with noise get classified as a vowel. Proper normalization methods can reduce this issue. When segmentation was performed on the unclean dataset and the resulting syllables were given, SVM outperformed KNN. This could be attributed to the fact that KNN is instance based and was trained on manually segmented phonemes. Hence all variations of phonemes were not captured due to limited data. The reason for better performance of SVM can be because the weights given to each output class in SVM were tweaked to give more importance to consonants that were misclassified often because of lesser data frames.

Table – V: Vowels and consonants (as phonemes) correctly classified

Method	Vowels (total 12)	Consonants (total 28)
CNN	6	13
SVM	9	11
KNN	8	19

Table – VI: Correctly classified vowels, consonants and syllables for Kannada Kali

Method	Vowels (total 6)	Consonants (total 14)	Syllables (total 26)
SVM	5	9	7
KNN	6	7	5

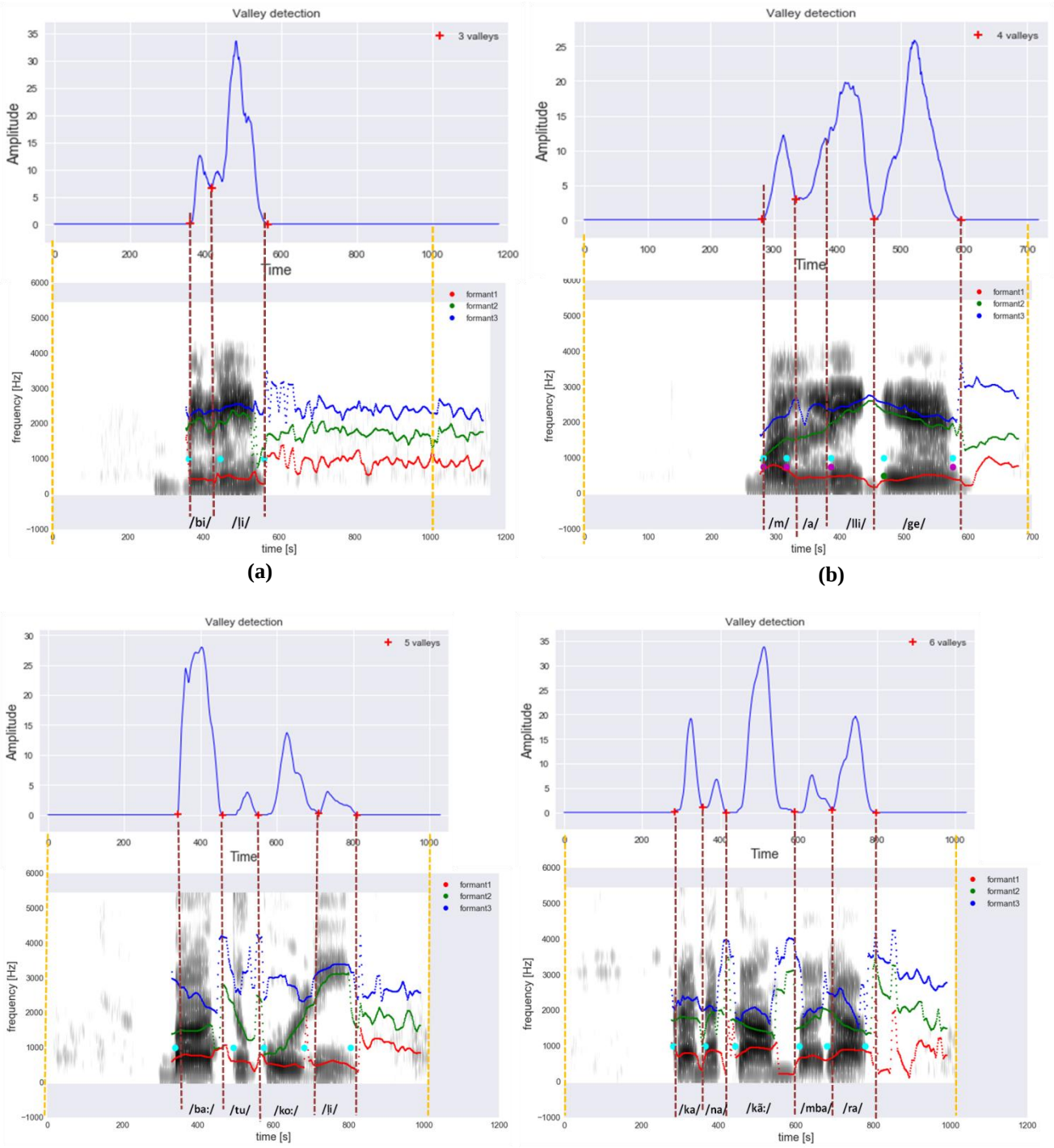


Fig. 5. Aligned syllable boundaries using Peak Valley Detection and SFCA method

V. CONCLUSION

In this paper, our research involves unsupervised segmentation of spoken Kannada words which is comparable with the well established peak-valley method and syllable classification under low resource conditions to aid mispronunciation detection. We believe that, with more experimentation, SFCA can be extended from syllable segmentation to determining phoneme boundaries which would make it very effective under zero resource conditions. Our classification study reveals that SVM performs marginally better than KNN for syllable classification when presented data from unsupervised segmentation. Since our segmentation approach is completely unsupervised and also, our classification uses minimal data, our approach can be extended to other Indian languages.

In future we intend to further determine the accuracy of our segmentation method with continuous speech data. We further plan to determine the validity of our syllable classification with more samples of audio.

REFERENCES

1. Al Hindi, A., Alsulaiman, M., Muhammad, G., & Al-Kahtani, S. (2014, November). Automatic pronunciation error detection of nonnative Arabic Speech. In 2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA) (pp. 190-197). IEEE.
2. Hu, W., Qian, Y., & Soong, F. K. (2013, August). A new DNN-based high quality pronunciation evaluation for computer-aided language learning (CALL). In Interspeech (pp. 1886-1890).
3. Hu, W., Qian, Y., & Soong, F. K. (2014, May). A DNN-based acoustic modeling of tonal language and its application to Mandarin pronunciation training. In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 3206-3210). IEEE.
4. Hu, W., Qian, Y., Soong, F. K., & Wang, Y. (2015). Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers. Speech Communication, 67, 154-166.
5. Liu, P., Yuen, K. W., Leung, W. K., & Meng, H. (2012, December). menunciate: Development of a computer-aided pronunciation training system on a cross-platform framework for mobile, speech-enabled application development. In 2012 8th International Symposium on Chinese Spoken Language Processing (pp. 170-173). IEEE.
6. Bhat, C., Srinivas, K. L., & Rao, P. (2010, December). Pronunciation scoring for Indian English learners using a phone recognition system. In Proceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia (pp. 135-139). ACM.
7. Duan, R., Kawahara, T., Dantsuji, M., & Zhang, J. (2017, March). Effective articulatory modeling for pronunciation error detection of L2 learner without non-native training data. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5815-5819). IEEE.
8. Qian, X., Meng, H., & Soong, F. (2016). A two-pass framework of mispronunciation detection and diagnosis for computer-aided pronunciation training. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 24(6), 1020-1028.
9. Laborde, V., Pellegrini, T., Fontan, L., Mauclair, J., Sahraoui, H., & Farinas, J. (2016). Pronunciation assessment of Japanese learners of French with GOP scores and phonetic information.
10. Li, W., Chen, N. F., Siniscalchi, S. M., & Lee, C. H. (2017, September). Improving Mispronunciation Detection for Non-Native Learners with Multisource Information and LSTM-Based Deep Models. In INTERSPEECH (pp. 2759-2763).
11. Franco, H., Neumeyer, L., Ramos, M., & Bratt, H. (1999). Automatic detection of phone-level mispronunciation for language learning. In Sixth European Conference on Speech Communication and Technology.
12. Witt, S. M., & Young, S. J. (2000). Phone-level pronunciation scoring and assessment for interactive language learning. Speech communication, 30(2-3), 95-108.
13. Witt, S. M., & Young, S. J. (2000). Phone-level pronunciation scoring and assessment for interactive language learning. Speech communication, 30(2-3), 95-108.
14. Arora, V., Lahiri, A., & Reetz, H. (2017). Phonological feature based mispronunciation detection and diagnosis using multi-task DNNs and active learning.
15. Rouhe, A., Karhila, R., Smit, P., & Kurimo, M. (2017). Reading Validation for Pronunciation Evaluation in the Digitala Project. In INTERSPEECH (pp. 2050-2051).
16. Hsu, Y. C., Yang, M. H., Hung, H. T., & Chen, B. (2016, January). Mispronunciation Detection Leveraging Maximum Performance Criterion Training of Acoustic Models and Decision Functions. In INTERSPEECH (pp. 2646-2650).
17. Lee, A., & Glass, J. (2013). Pronunciation assessment via a comparison-based system. In Speech and Language Technology in Education.
18. Lee, A., Zhang, Y., & Glass, J. (2013, May). Mispronunciation detection via dynamic time warping on deep belief network-based posteriorgrams. In 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 8227-8231). IEEE.
19. Kumar, S., Phadikar, S., & Majumder, K. (2016, December). Modified segmentation algorithm based on short term energy & zero crossing rate for maithili speech signal. In 2016 International Conference on Accessibility to Digital World (ICADW) (pp. 169-172). IEEE.
20. Zahner, K., Kember, H., & Braun, B. (2017). Mind the peak: When museum is temporarily understood as musical in Australian English. In Interspeech 2017 (pp. 1223-1227).
21. Chithra, P. L., & Aparna, R. (2015). Performance analysis of windowing techniques in automatic speech signal segmentation. Indian Journal of Science and Technology, 8(29), 1.
22. Yarra, C., Deshmukh, O. D., & Ghosh, P. K. (2016). A mode-shape classification technique for robust speech rate estimation and syllable nuclei detection. Speech Communication, 78, 62-71.

AUTHORS PROFILE

Savitha Murthy is a Research Scholar currently pursuing her PhD under the guidance of Dr. Dinkar Sitaram at P.E.S University, Bangalore in the department of Computer Science and Engineering. She has teaching experience of 6 years and industry experience of 4 years. Her area of research is in Automatic Speech Recognition for low resource languages. She has authored 8 publications.

Pragnya Suresh is currently pursuing a Bachelor's degree in Computer Science Engineering from PES University, Bangalore. Her areas of interest include Machine Learning, Natural Language processing and Cloud Computing. She is currently working on multimodal emotion recognition in a dyadic conversation. Her other projects in the domain of Natural Language processing include "Automated Trouble Ticket Routing"-a model for online shopping portals to automatically route reviews and complaints from social media to enhance customer support that also employs methods to help resolve the customer's complaint, thus reducing manual work in the customer service domain. She has worked on projects involving containerizing applications to be deployed in a microservice architecture.

Preet Shah is currently pursuing a Bachelor's degree in Computer Science Engineering with ML and AI specialization from PES University. He is interested in algorithms and art. His current work includes emotion recognition in dyadic conversations using multiple modes - text, audio and video. He has published a paper titled "Faster Convergence to N-Queens Problem using Reinforcement Learning" in the MMLA 2019 conference. His other works include automated routing of product reviews and issues posted on social media to particular domains using Named Entity Recognition and automatically asking for missing fields required to solve the issue, improved query search technique based on Ukkonen suffix trees and designing an AI agent to play Nine Men's Morris.

Dr. Dinkar Sitaram completed his Ph.D in Computer Science from the University of Wisconsin Madison. His research interests include scheduling and sustainable resource allocation policies for hybrid clouds and cloud computing, as well as speech recognition and intelligent big data systems. He has authored 2 books – 'Moving to the Cloud – Developing Apps **in the New World** of Cloud Computing' published by American Elsevier; and 'Multimedia Servers' published by Morgan Kaufman. He has authored over 30 patents and 45 publications. He was awarded the IBM Corporate Innovation Award. He has served as CTO of Novell Software, as CTO of Andiamo Systems and also as CTO of STSD (R&D Wing of HP, India) prior to joining PES University.